The first commit to tinygrad was October 17, 2020. It's been almost three years since we raised money. The company is 6 people now. The codebase is 18,935 lines not including tests.

I have spent 5 years of my life working on 18,935 lines, and now many others have put years in as well. And there's probably 5 more years to go. But this is the right process to compete with NVIDIA.

Only a fool begins by taping out a chip; it's expensive and not the hard part. Once you have a fully sovereign software stack capable of training SOTA models, the chip is so easy. Note that AMD, Amazon, Tesla, and Groq have taped out fine chips, but only Google and NVIDIA chips have ever been seriously used for training. Because they have the software.

We are finally beginning to tackle LLVM removal, making tinygrad have 0 dependencies (except pure Python) to drive AMD GPUs. We have a frontend, we have a graph compiler, we have runtimes, and we have drivers. This is no longer a toy project, it outperforms PyTorch on many workloads. When this is finished and cleaned up, it'll be about 20,000 lines. And that's completely it.

I think a lot of how software is thought about is wrong. All codebases have workarounds for issues in other parts of the codebase. Sometimes you are lucky and these workarounds are clear, but many times they are so deep and structural that you'll never see them reading the code line by line. I think this is so bad that 98% of lines of software are basically this in some way shape or form. tinygrad is following the Elon process for software.

> *Make the requirements less dumb. The best part is no part.*

Most requirements in software exist to maintain compatibility with other abstractions. Let's look at an LLM server. The real requirement is that it provide an OpenAI compatible API for quickly running LLMs on GPUs. But when you look at these codebases and what they depend on, they are collectively millions and millions of lines. tinygrad is 1000x smaller.

It's because each piece of code in that stack isn't focused on the goal. It's focused on the other pieces of code. I believe this is the same dysfunction that exists in organizations too.

The tiny corp is a company, but like how fancy chefs will deconstruct dishes, the tiny corp is a deconstructed company. We have almost nothing private, it’s a Discord and GitHub. To fund the operation, we have a computer sales division that makes about $2M revenue a year.

We also have a contract with AMD to get MI350X on MLPerf for Llama 405B training. This was negotiated mostly in public on Twitter.

People get hired by contributing to the repo. It’s a very self directed job, with one meeting a week and a goal of making tinygrad better.

Our mission is to **commoditize the petaflop**.